**PySpark**

**Mini Project**

**Hockey Analysis**

**Table of Contents**

[**Introduction 3**](#_heading=h.1fob9te)

[1.1](#_heading=h.2jxsxqh) Setup Checklist for Mini Project 3

[1.2](#_heading=h.z337ya) Instructions 3

[**2**](#_heading=h.tyjcwt) **Problem Statement 4**

[2.1](#_heading=h.tyjcwt) Objective 4

[2.2](#_heading=h.3dy6vkm) Abstract of the project 4

[2.3](#_heading=h.1t3h5sf) Technology use 4

[**3**](#_heading=h.4d34og8) **Implementation in PYSPARK LOT 5**

[3.1](#_heading=h.3j2qqm3) Summary of the functionality to be built 5

[3.2](#_heading=h.17dp8vu) Guidelines on the functionality to be built 5

[**4**](#_heading=h.1y810tw) **Reports to be built 11**

[**5**](#_heading=h.44sinio) **Appendix (source Files) 13**

# Introduction

This document outlines a mini project for the Pyspark LOT. The project is to develop Analysis & Reporting System of Hockey Game.This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the PySpark LOT.

* 1. **Setup Checklist for Mini Project**

**Minimum System Requirements**

* Hardware:
  + - Intel Pentium 90 or higher (P166 recommended)
    - Microsoft Windows 2010 or above.
    - Memory: 4GB of RAM (4GB or more recommended)
* Software:
  + - Anaconda
    - Python
    - IDE – Pycharm/Jupyter Notebook
    - Pyspark
    - Internet Explorer 10.0 or higher

NOTE: Anaconda will install Python. Other tools like Jupyter notebook, Spyder can be installed through Anaconda as well.

* 1. **Instructions**
* The code modules in the mini project should follow all the coding standards.
* Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **MiniProject**. Store your Project here.
* You can refer to your course material.
* You may also look up the help provided in the Pyspark docs and documentation provided in the respective tools.

# Problem Statement

* 1. **Objective**

Development of a Analysis & Reporting System of Hockey Game

* 1. **Abstract of the project**

The project is to develop **Analysis & Reporting System of Hockey Game.**

* 1. **Technology use**
* Anaconda
* Python
* IDE – Pycharm/Jupyter Notebook
* Pyspark

# Implementation in PYSPARK LOT

* 1. **Summary of the functionality to be built**

The participants need to develop the **Analysis & Reporting of Hockey game** by building the functionality incrementally in each of the course modules of PYSPARK LOT.

* 1. **Guidelines on the functionality to be built**

**Project flow**

Heterogeneous Sources i.e. operational data (Flat files)

EXTRACT

Load

Transform

In-memory

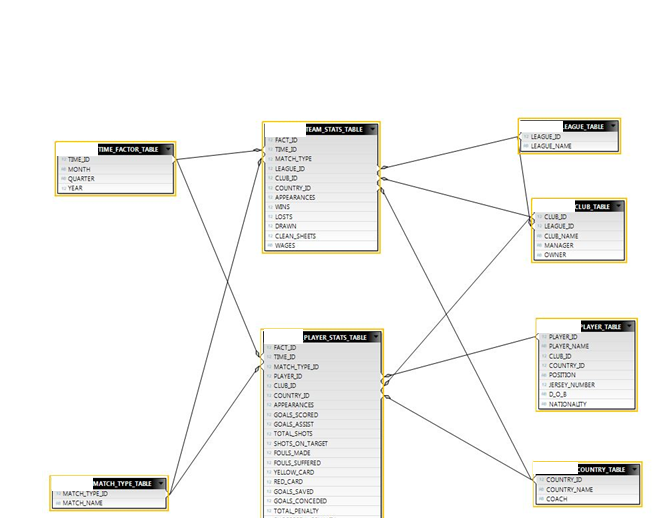
Dimension Model using PySpark

REPORTS

(CSV)

* The participants are expected to create a star/snowflake schema model based on the OLTP source files present in Appendix
* **The Source Files Are:**
* **Source File 1**
  + Year
  + Match
  + Player Name
  + Club Name
  + Country Name
  + Appearances
  + Goals scored
  + Goals assist
  + Total shots
  + Shots on target
  + Fouls made
  + Fouls suffered
  + Yellow card
  + Red card
  + Goals saved
  + Goals conceded
  + Total penalty
  + Successful penalty
  + Salary
* **Source File 2**
  + Year
  + Match Name
  + League Name
  + Club Name
  + Country Name
  + Appearances
  + Wins
  + Losts
  + Drawn
  + Clean sheets
  + Net Worth
* **Date Dimension**
  + Date Key
  + Full Date
  + Day of week
  + Day Num in Month
  + Day Num Overall
  + Day Name
  + Day Abbrevation
  + Week Day Flag
  + Week Number in Year
  + Week Number overall
  + Week Begin Date
  + Week Beging Date Key
  + Month
  + Month Number overall
  + Month Name
  + Month Abbrevation
  + Querter
  + Year
  + Year Month
  + Fiscal Month
  + Fiscal Quarter
  + Fiscal Year
  + Last Day in Month Flag
  + Same day year ago date

In-Memory dimensional model relationship for reference purpose.

****

**The Dimension tables are**

* + **League**
    - League ID
    - League Name
  + **Club**
    - Club\_id
    - League\_id
    - Club\_Name
    - Manager
    - Owner
  + **Country**
    - Country\_id
    - Country\_Name
    - Coach
  + **Match Type**
    - Match\_type\_id
    - Match\_Name
  + **Player**
    - Player id
    - player name
    - club\_id
    - Country\_id
    - position
    - Jersey number
    - D.O.B
    - nationality
  + **Time**
    - Time\_id
    - Year
    - Quarter
    - Month

**Fact Tables are**

* **Fact Player Statistics**
  + Fact\_id
  + Time\_id
  + Match\_type\_id
  + Player\_id
  + Club\_id
  + Country\_id
  + Appearances
  + Goals scored
  + Goals assist
  + Total shots
  + Shots on target
  + Fouls made
  + Fouls suffered
  + Yellow card
  + Red card
  + Goals saved
  + Goals conceded
  + Total penalty
  + Successful penalty
  + Salary
* **Fact Team Statistics**
  + Fact\_id
  + Time\_id
  + Match\_type\_id
  + League\_id
  + Club\_id
  + Country\_id
  + Appearances
  + Wins
  + Lost
  + Drawn
  + Clean sheets
  + Net Worth

**Note:** The datatype/length for the Dimension/Fact table attribues can be changed as required. Additional fields can be added, if required.

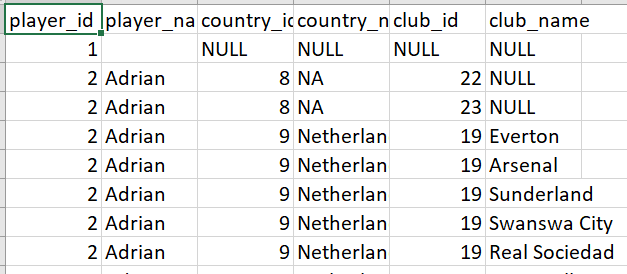
**Data Transformation for data warehouse:**

1. Load the Data into Dimension tables using the CSV files provided.
2. Load the Data into Fact tables using the CSV files provided.
3. Populate Dimension tables before fact table
4. Dimension table should have unique values with unique system generated IDs. Rank() function can be used to generate the unique ID (sequence). Same for Fact ID
5. Other ID columns in Fact table example in Players Fact table – Match\_type\_id, player\_id, club\_id etc. column should be populated with id columns values from corresponding dimension table

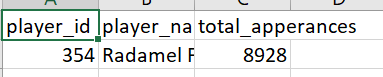
# Reports to be built

Following reports to be created. Create CSV file output for each report. Reports to be generated from the Fact and Dimension tables

* Player report with country and club they played for



* Player report who has maximum appearances in International matches



* Find top 10 goal scorers of 2010



* Find top 5 players who got red card



* Find the top 5 successful clubs



# Appendix (source Files)

